

SSK2_INH17G_Klausur_1_S2-90 - BUN - 15.08.2018

Schriftliche Klausur 15.08.2018 – 90 Minuten
Fach / Lernfeld SSK2
Semester / Klasse SS 18 / INH17G

Hinweise:

1. Hilfsmittel: Unity und Visual Studio in den vorliegenden Versionen.
2. Der zu erstellende Code muss nicht zwingend lauffähig sein, falls Anschlussstellen zwischen den Aufgaben nicht hergestellt wurden. Die Aufgaben werden isoliert für sich bewertet; geforderter Code zum Aufrufen potenziell nicht vorliegender Daten muss dazu aber existieren. Berücksichtigen Sie Best Practices in Unity und C#.
3. Exportieren Sie Ihre Ergebnisse als unitypackage mit dem Namen:
SSK2_KLASSE_NACHNAME_VORNAME_DATUM.unitypackage

Schriftliche Klausur 15.08.2018 – 90 Minuten
Fach / Lernfeld SSK2
Semester / Klasse SS 18 / INH17G

Ausgangslage

Für den Einsatz in einem mobile Game sollen Sie den Protoyp eines Kartenspiels (TCG) entwickeln.

20/100 Punkte

1. Erstellen Sie unter Berücksichtigung bekannter Struktur-Paradigmen eine geeignete Struktur von (abstrakten) Datentypen (Klassen, Structs oder Interfaces), um Karten/cards mit Ableitungen und konkreten konkreten Implementationen: Character, Spell und Ability abbilden zu können.

Alle Karten haben mindestens die Eigenschaften:

- value, enthält:
 - name
 - id
 - basecost
 - quality (bronze, silver, gold)
- Use(Ability ability)

Character

- int health
- CharacterType (friend, foe)
- Use / individuelle u. konkrete Implementation in sinnvollem Kontext

Spell

- Ability ability
- ApplyEffect(Card targetcard)
- Use / individuelle u. konkrete Implementation in sinnvollem Kontext

Ability

- Use / individuelle u. konkrete Implementation in sinnvollem Kontext

Stellen Sie die jeweiligen Eigenschaften mit dafür geeigneten Datentypen dar, die Sie dem typischen Sinn der Bezeichnung (des Namen) eines Feldes entnehmen können.

10/100 Punkte

2. Das Feld value soll in Form eines zusammengesetzten Datentypen abgebildet werden, welcher speichereffizienter ist, als eine Klasse.

Beim Ausführen der ToString() Methode der Basisklasse sollen per string.Format() die Felder des value Datentyps speichereffizient ausgegeben werden.

Der Datentyp Ability simuliert Aktionen, welche beim Ausführen der Use() Methode als String per Debug.Log ausgegeben werden.

3. 35/100 Punkte

Implementieren Sie konkrete Varianten von Abilities, welche:

- Move: eine Bewegung ausführt/simuliert

Schriftliche Klausur 15.08.2018 – 90 Minuten
Fach / Lernfeld SSK2
Semester / Klasse SS 18 / INH17G

- Damage u. HealUnit: einen Wert enthält, der auf einer Einheit (Character) als Schaden oder Bonus angewendet werden kann und die Verwendung spezifiziert
- Überlegen Sie sich einen sinnvollen Einsatz der Methode Use() der Basisklasse

Implementieren Sie eine konkrete Variante eines Spell, welcher

- eine konkrete Ability besitzt
- die Effekte dieser Fähigkeit mit der Methode ApplyEffect() an eine Zielkarte weitergeben kann
- überlegen Sie sich einen sinnvollen Einsatz der Methode Use() der Basisklasse

Implementieren Sie eine konkrete Variante eines Character, welcher

- einen Wert für health besitzt
- mit der Methode ReceiveEffect eine Ability entgegennehmen kann und den darin enthaltenen Wert entweder positiv oder negativ auf sein Feld health anwendet
- Abilities besitzt (z.B. Move) welche auf sich selbst angewendet werden können und ebenfalls auch auf andere Karten anwendbare Fähigkeiten, wie z.B. einen Spell
- Überlegen Sie sich einen sinnvollen Einsatz der Methode Use() der Basisklasse und erzeugen Methoden, damit die Karte ihre unterschiedlichen Fähigkeiten einsetzen kann.

5/100 Punkte

4. Erstellen Sie ein Asset, welches eine Liste von Karten zur EditTime persistent halten kann.

Schaffen Sie ebenfalls die Möglichkeit, damit Characters, Spells und Abilities als Asset manuell vorkonfiguriert werden können.

30/100 Punkte

5. Erstellen Sie einen TestController als MonoBehaviour, welche nun auf ButtonClicks eine öffentlich konfigurierbare Anzahl von Characters (mit Abilities) und Spells erstellt, und diese dann miteinander/gegeneinander ausführen kann.

Erzeugen Sie zwei Prefabs als Primitives, welche Character und Spell visualisieren. Diese sollen einen MonoBehaviour als Sockel besitzen, welcher entweder einen konkreten Instanz Character oder Spell enthält.

Folgend soll nun per Button Klick

- aus einer Liste zwei zufällige Karten ausgewählt werden,
- die GameObjecte dazu instanziiert werden.
- Falls es sich um zwei Karten handelt, so soll eine Karte ihre Ability auf die andere Karte einsetzen; handelt es sich um eine Karte und einen Spell, so soll der Spell auf die Karte angewendet werden; handelt es sich um zwei Spells, so wird ein erneutes Paar bestimmt

Diese Unterscheidung soll per Ternary-Ausdruck und über die Auswertung des Datentyps geschehen.